Rule WLM705: Staging data sets not efficiently used, DASD-only log stream

Finding: The SMF Type 88 data showed that staging data sets were not efficiently used for a DASD-only[1] log stream.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of your computer system. The level of impact depends on the applications using the log stream, and the extent to which log stream delays effects the performance of these applications.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data.

Discussion: The system logger is an MVS component that allows an application to log data from a sysplex. You can log data from one system or from multiple systems across the sysplex.

Please refer to Rule WLM701 for more general information about the MVS system logger.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[2], where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is “hardened” for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only log streams. The main difference between the two types of log streams is the storage medium that the system logger uses to hold interim log data:

- With a coupling facility log stream, interim storage is contained in coupling facility list structures. The system logger duplexes the log stream to either (1) MVS data space areas associated with the system logger address space or (2) staging data sets, depending on whether the coupling facility is failure-independent.

- With a DASD-only log stream, interim storage is contained in local storage buffers on the system (as MVS data space areas associated with

[1] DASD-only log streams are supported beginning with OS/390 Version 2 Release 4.

[2] Interim storage is sometimes referred to as “primary” storage.

the system logger address space). With a DASD-only log stream the system logger duplexes the log stream to staging data sets

Interim storage normally is “offloaded” to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled. For log streams defined in coupling facility list structures, the parameters apply to the coupling facility structures[3].

For log streams defined as DASD-only, these parameters apply to the log stream staging data set, as follows:

- When the staging data set is filled to the **high offload threshold** point or beyond, the system logger begins offloading data from the staging data set to the DASD log stream data sets. For example, if the HIGHOFFLOAD parameter is specified as 80% (this is the default value), the system logger normally would begin offloading log stream data from the staging data set to DASD log data sets when 80% or more of the staging data set has been used.

- The **low offload threshold** is the point in the staging data set, as a percent of space consumed, where the system logger stops offloading log data in the staging data set to log stream DASD data sets. The default LOWOFFLOAD parameter value is 0%, indicating that the system logger will offload all the log stream to DASD log data sets once offloading has commenced.

Once log stream data has been offloaded, the MVS system logger releases the storage in the staging data sets, so the space in the staging data sets can be used to hold new log blocks. From an application point of view, the actual location of the log data in the log stream is transparent.

Applications using system logger services (such as CICS/Transaction Server for OS/390) often manage the system log by deleting records for completed units of work during activity keypoint processing (this is also called log-tail deletion). The number of bytes deleted from the system log after writing to offload data sets should be very low. Unnecessary overhead is incurred when data is moved to the offload data sets, only to be later deleted. With an appropriately sized log stream, the system log data remains in interim storage, and the overhead of data spilling to DASD simply to be deleted later is avoided.

[3] The parameters will also apply to staging data sets if the log stream is duplexed to staging data sets. Problems with staging data set threshold being encountered are analyzed in Rule WLM705.

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file.

CPExpert computes the percent of ineffective use of staging data sets (PCTLOCST) by applying the following algorithm to DASD-only log streams:

$$PCTLOCST = \frac{SMF88SAB}{SMF88SIB + SMF88SAB}$$

where

SMF88SAB	=	Bytes deleted after being offloaded
SMF88SIB	=	Bytes deleted before being offloaded

CPExpert compares the computed PCTLOCST with the **PCTLOCST** guidance variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM704 when the percent ineffective use of use of interim storage exceeds the value specified by the **PCTLOCST** guidance variable.

The default value for the **PCTLOCST** guidance variable is 0, indicating that CPExpert should produce Rule WLM705 whenever DASD staging data set use was not effective.

Suggestion: The delete after offload percent is a key indicator that log tail deletion is not working as effectively as it should. If significant values appear in this percent, you should consider the following alternatives:

- For CICS/TS, verify that SYSLOG=KEEP is not specified as a System Initialization Table (SIT) parameter (this suggestion applies only to CICS/TS Release 1.1, as the SYSLOG keyword was made obsolete with CICS/TS Release 1.2). The SYSLOG=KEEP option inhibits CICS from deleting data from the system log, even though the data is no longer needed. IBM strongly recommends that the SYSLOG=NOKEEP option be used, and the SYSLOG keyword was removed from the SIT with CICS/TS Release 1.2.

- Verify that there are not any long running transactions making recoverable updates without syncpointing

- Consider increasing the HIGHOFFLOAD threshold value.

- For CICS/TS, examine the System Initialization Table (SIT) values for this region, and determine whether AKPFREQ is zero or is too high. With a CICS/ESA 4.1 region (or earlier), the AKPFREQ parameter specifies the number of consecutive blocks written to the system log data set. However, with CICS/TS for OS/390, the AKPFREQ parameter represents the number of write operations (log records) by CICS log manager to the log stream buffer before an activity keypoint is taken.

- If AKPFREQ=0, CICS cannot perform log tail deletion until shutdown, by which time the system log will have spilled to secondary storage. This situation would elongate shutdown and cause unnecessary overhead.

- The AKPFREQ parameter has a significant impact on the size of system logger primary (interim) storage, affecting the log tail management that takes place during activity keypoint (AKP) processing. During AKP processing, the system logger deletes records that are no longer of interest to CICS and moves records to DFHSHUNT for those tasks that did write any log records within the last AKP interval.

- In an MRO environment, the keypoint program uses an appreciable amount of CPU capacity in processing persisting units of work such as those relating to mirror transactions waiting to process an implicit forget. This is exacerbated when the AKPFREQ value is low. An optimum setting of AKPFREQ allows many of these persistent units of work to complete during normal transaction processing activity. This minimizes the CPU processing used by the keypoint program. IBM suggests that you exercise caution in reducing the value of AKPFREQ below the default value.

- Consider increasing the size of the DASD staging data sets.

- You can alter CPExpert's analysis by modifying the **PCTLOCST** guidance variable in USOURCE(WLMGUIDE).

Reference: *CICS/TS Release 1.1 Performance Guide*:
Section 4.6.1 (Monitoring the logger environment).
Section 4.6.7: Activity keypoint frequency (AKPFREQ).

CICS/TS Release 1.2 Performance Guide:
Section 4.6.2: Monitoring the logger environment.
Section 4.6.7: Activity keypoint frequency (AKPFREQ).

CICS/TS Release 1.3 Performance Guide:
Section 4.8.2: Monitoring the logger environment.
Section 4.8.7: Activity keypoint frequency (AKPFREQ).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 22:
Monitoring the logger environment.
Activity keypoint frequency (AKPFREQ).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 22:
Monitoring the logger environment.
Activity keypoint frequency (AKPFREQ).